A FRAMEWORK FOR GENERIC SPATIAL SEARCH IN 3D POINT CLOUDS

J. Otepka, G. Mandlburger, W. Karel, B. Wöhrer, C. Ressl, N. Pfeifer

TU Wien, Department of Geodesy and Geoinformation, Vienna, Austria -
johannes.otepka@geo.tuwien.ac.at

Commission II, WG II/3

KEY WORDS: 3D point cloud, neighbourhood definition, k nearest neighbour search, fixed distance search, cell based sampling, incremental spatial search

ABSTRACT:

Modern data acquisition with active or passive photogrammetric imaging techniques generally results in 3D point clouds. Depending on the acquisition or processing method, the spacing of the individual points is either uniform or irregular. In the latter case, the neighbourhood definition like for digital images (4- or 8-neighbourhood, etc.) cannot be applied. Instead, analysis requires a local point neighbourhood. The local point neighbourhood with conventional k-nearest neighbour or fixed distance searches often produce sub-optimal results suffering from the inhomogeneous point distribution. In this article, we generalize the neighbourhood definition and present a generic spatial search framework which explicitly deals with arbitrary point patterns and aims at optimizing local point selection for specific processing tasks like interpolation, surface normal estimation and point feature extraction, spatial segmentation, and such like. The framework provides atomic 2D and 3D search strategies, (i) k-nearest neighbour, (ii) region query, (iii) cell based selection, and (iv) quadrant/octant based selection. It allows to freely combine the individual strategies to form complex, conditional search queries as well as specifically tailored point sub-selection. The benefits of such a comprehensive neighbourhood search approach are showcased for feature extraction and surface interpolation of irregularly distributed points.

1. INTRODUCTION

Point clouds are sets of 3D points (xyz), which may potentially feature additional attributes. Following the definition of (Otepka et al., 2013) not all points do necessarily have the same attributes. As the point cloud constitutes an unordered set, no explicit spatial topology exists in general. This is different to triangulations of point clouds or grids of 3D points. Today, 3D point clouds are commonly accepted as a prime product of both active and passive photogrammetry. Also data acquired with other measurement methods, e.g. multi beam echo sounding, is being treated as point cloud (Held and Schneider von Deimling, 2019). While point clouds were long regarded as an intermediate product, e.g., for the derivation of digital surface and terrain models (DSM/DTM) or 3D models (e.g. in building information modeling, BIM), they represent a stand-alone product today and play an important role in visualization, data processing, and GIS. Accordingly, also data structures for point clouds gained importance in research. A fundamental step for processing point clouds is the extraction of information in the vicinity of a point. Such a point can be, but is not required to be, part of the given point cloud. Finding neighbours is therefore of paramount importance in point cloud processing.

Dense Image Matching applied to aerial nadir images, for instance, produces a homogeneous point pattern as 3D points are generated for every image pixel. In contrast, point clouds obtained from feature based matching exhibit an irregular structure. The same applies to laser scanners employing oscillating mirrors or rotating bevelled scan wedges (Palmer scanners). In this case the center part of the swath often exhibits uniform point spacing along and across track, whereas the strip boundaries feature a much higher point density in one direction. Also terrestrial laser scanning (TLS) features a decreasing point density with increasing distance from the scanner. In a block of TLS scans, the combined point cloud exhibits multiple high point density clusters with complex patterns of decreased density in between.

Localized processing of point clouds imperatively requires an appropriate neighbourhood definition. The results of feature extraction and interpolation tasks depend on the algorithm and its parameters, but also on the neighbourhood definition itself. A few algorithms imply a certain neighbourhood definition, as e.g. the natural neighbour interpolation (Bobach and Umlauf, 2006) or the Delaunay triangulation (de Berg et al., 2000). But for most algorithms, the spatial neighbourhood definition can be defined independently which is an important concept for designing a generic spatial search framework. Nevertheless, processing algorithms may presume certain constraints for the spatial search. E.g., moving planes interpolation or local normal estimation require at least three neighbouring points for successful computation. Those constraints can be formalized within the proposed framework for spatial search in point clouds.

The most common neighbour definitions can be categorized into (i) nearest neighbour, (ii) region queries or (iii) a combination of both. We concentrate on these definitions, because they cover a wide variety of neighbourhoods that were reported in literature. Definitions not covered by the above categories include (i) neighbourhoods that require some optimization (e.g. barycentric neighbourhood, slope adaptive neighbourhood) or (ii) more complex steps of preprocessing (e.g. surface reconstruction for measuring distance along this surface). For data with nearly isotropic and homogeneous point distributions it is largely irrelevant if either nearest neighbour or region queries are applied (assuming matching query parameters). Nevertheless, some processing algorithms require certain properties of the neighbour definition to work correctly. E.g. the result of region growing is only independent of seed points and data order if symmetrical coherence predicates and symmetrical

https://doi.org/10.5194/isprs-annals-V-2-2021-35-2021 |

neighbour definitions are used. Since k-nearest neighbour queries (kNN) are non-symmetrical in general, symmetric region queries are preferred for those applications. 'Symmetry' means that points are mutually neighbours under a constant neighbourhood definition. On the other hand, k nearest neighbour queries can compensate for varying point densities resulting in a constant count ($= k$) of neighbours which is why it is used in many processing strategies. However, in some situations a constant neighbour count is disadvantageous, e.g. for interpolating high quality surface models. There, regions with larger data gaps should not be bridged by the interpolation since the result might be erroneous. Models that do not close gaps are often preferred (at least as intermediate result) for further analyses. The situation gets even more difficult, if the point distribution is inhomogeneous. If the spacing of LiDAR points in scan line direction substantially differs from the spacing between scan lines, the k nearest neighbours may result from one scan line only. This can lead to extrapolation and artefacts in the results. On the other hand, a fixed distance region definition either leads to data gaps or unnecessary smoothing if the distance is chosen too small or too large.

The proposed framework allows applying and combining different spatial search strategies to avoid extrapolation in a very generic way that is separated from the actually processing algorithm. Flexibility, however, does not come without costs. Obviously, a highly optimised tool which combines a specific spatial query and processing algorithm within an atomic unit, will outperform a generic framework. On the other hand, such an optimization might be usable for certain sensor data only or it might require additional pre- or post-processing steps reducing or even consuming a possible performance advantage. Based on the enormous measurement rates ($> 1\,\mathrm{Mhz}$) of modern laser scanners, as well as, Dense Image Matching of images using today's aerial cameras, tremendous point clouds can be produced. While the authors acknowledge the importance of processing performance, the focus of this paper is not on efficient implementation, but rather on defining a generic framework for spatial search in point clouds.

The specific contributions of this article are:

- Definitions of atomic neighbourhoods in a common nomenclature. 'Atomic' means that such a neighbourhood definition is of a specific type, which cannot be subdivided further.
- Introduction of a 'cell-based' neighbourhood.
- A framework to combine the atomic neighbourhoods using union or intersection operations.
- A new incremental neighbourhood definition, which gradually expands an initial region until a condition is met or the maximum region size is reached.

The remainder of this article is structured as follows: Section 1.1 briefly reviews related literature. Section 2 describes the basic neighbourhood atoms and the framework for combining the atoms, as well as, the data sets used for testing the approach. The results of the different neighbourhood definitions applied to the test data sets are presented and discussed in Section 3. The article ends with a summary of the main findings and conclusions in Section 4.

1.1 Related work

An overview on neighbourhoods used in point cloud processing is given in (Otepka et al., 2013). The neighbourhoods k-nearest neighbour and fixed distance search are described there by geometric definitions. Also (Filin and Pfeifer, 2005) review different neighbourhoods defined in point clouds and develop a 2-step approach to define a neighbourhood of a point, which is specifically tailored to the application of point cloud segmentation. Frequently, neighbourhood at different spatial scales is applied, in order to extract multi-scale features at each point (Weinmann et al., 2015). In classification tasks it is possible to either use machine learning strategies and features at different scales directly or to locally select the most relevant scale using entropy analyses. Therefore, a generic spatial search framework needs to support adaptive neighbourhood definitions as well. From a conceptional point of view adaptive neighbourhood scales can be realised by either incrementally increasing the spatial queries or, by retrieving the maximum neighbourhood first and then sub-selecting the relevant data. Which of the two methods result in higher performance depend on several parameters and the data itself. Nevertheless, the latter strategy can only be utilized, if a maximum neighbourhood scale is given. For situations where this is not the case, so called *all nearest neighbour* algorithms can be applied. (Sankaranarayanan et al., 2007) presented an efficient out-of-core *all nearest neighbour* algorithm which can handle huge point clouds that do not fit into memory.

Neighbourhood definitions based on 2D or 2.5D surface representations like Delaunay triangulations can effectively prevent extrapolation. Assuming a point lies within the convex hull of the triangulation, at least one triangular face can be found that contains or touches the point (Devillers et al., 2001). Hence, it is straightforward to query neighbouring points that surround the current query point, i.e. the convex hull of the neighbours itself contains the query point. The downside of a surface representation is the algorithmic and computational complexity, especially for huge and complex 3d point clouds. (Isenburg et al., 2006) presented a streaming Delaunay triangulation that is capable of triangulating huge point clouds that are many times larger than the available memory. It makes use of the fact that point cloud data always have some sort of spatial order. Based on the Delaunay criterion and the given spatial order, the algorithm only maintains changing parts of the triangulation in memory when streaming through the data. Finalised triangles are dropped from the triangulation and further processed by, e.g., an interpolation module. This concept, however, has two major drawbacks which make it difficult to apply within a generic spatial framework. First, the triangle stream does not correspond with the point cloud stream or any raster tile structure and therefore, it is always necessary to sort or post-process the outcomes for the final result. Secondly, the triangulation is conceptual a 2D structure and thus, inappropriate for complex 3D scenes. Nevertheless, it would be possible to extend the algorithm to a 3D tetrahedral triangulation.

In the early days of Laser Scanning computers and processing tools were often overburdened by the sheer amount of 3D data which is why simple sampling strategies like processing only every n$^{\text{th}}$ point were regularly utilized. A large variety of sampling methods have been developed which are commonly in use for increasing the processing performance e.g. (Schaer and Skaloud, 2010), (Blaszczak-Bak et al., 2020), as well as, for homogenizing the data distribution (Puttonen et al., 2013). If point clouds show a large inhomogeneity it is difficult to select appropriate processing parameters and a suitable neighbourhood definition. Therefore, many interactive point cloud processing software packages provide different sampling strategies to overcome those issues (CloudCompare Development Team, 2021), (MeshLab Development Team, 2021). In certain situations the processing is not only faster but even better results can be achieved if only a well selected sub-sampling point set is processed. (Glira et al., 2015) present and evaluate different sampling strategies for correspondences selection in ICP to improve the estimation of orientation parameters.

Classification and segmentation tasks often rely on features that are extracted from the point cloud (Niemeyer et al., 2012). From the well-known 3D structure tensor in combination with the principal component analysis (PCA) a variety of valuable features can be computed, such as, the local normal vector, linearity, planarity, etc. which reflect the shape of the current neighbourhood (Weinmann et al., 2015). In case the points are arranged along a linear structure the linearity measure will be high, whereas high planarity values result if the points are distributed along two axes. In case of inhomogeneous data distributions the neighbourhood definition must be carefully chosen, otherwise those features only reflect the data distribution rather than the captured object. The proposed spatial search framework provides a large variety of sampling and neighbourhood definitions that even difficult data distributions can be reasonably processed.

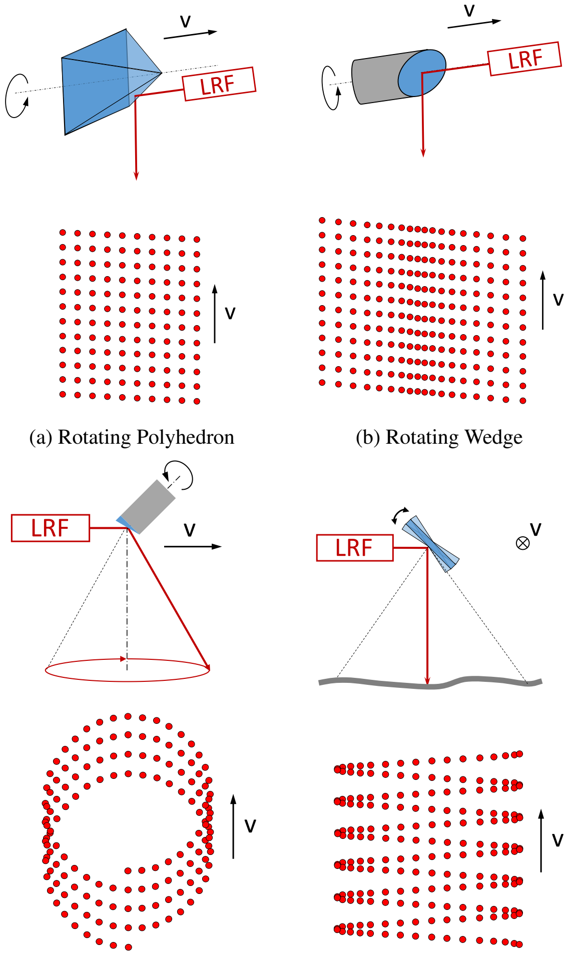
2. MATERIALS AND METHODS

2.1 Datasets

To demonstrate the flexibility of the proposed framework, four sample data sets from four different LiDAR sensors using different scan patterns (see Fig.1) were selected. The data properties range from very homogeneous in point density and point distribution to pronounced inhomogeneities of point distances along and across the scan lines. To exhibit the scanner pattern as clearly as possible, the data represent a single strip or single channel only. As it will be shown, computational artefacts are highly correlated with scan patterns. In real-world projects, the situation is typically more complex due to multi channel or multi strip coverage. However, the inhomogeneous point distribution will never be completely resolved and in some regions the inhomogeneity can even increase. Details of the data sets are given in Table 1 and height color coded 3D views are presented in Fig. 2.

To quantify the homogeneity of the data sets, the inhomogeneity factor f introduced by (Pfeifer et al., 2021) is used. For each point, the nearest neighbour per quadrant is computed. The distances to the four neighbours are sorted in ascending order and named $d_1 - d_4$. f is then defined as $f = d_4/d_1$. For scan patterns featuring an equal point spacing within and across the scan lines, f is close to 1. If the point density is, e.g., twice as high in scan line direction, a factor of 2 will yield. In (Pfeifer et al., 2021), this measure is applied to data which is expected to lie in a plane, whereas the data studied here is distributed in 3D. In case of (nearly) vertical surfaces an adapted selection strategy is required. As the test data sets investigated in the study at hand only exhibit a low number of points on vertical surfaces, this issue can simply be neglected, by selecting the median f from all points.

In the four test sets, the number of points span from 61K to around 80k. The average point density ranges from 0.3-3104 pts/m^2 and the median of f is between 1.2 and 14.8. Due to the high variation of inhomogeneity factors, the data sets are an ideal test case for the proposed framework.



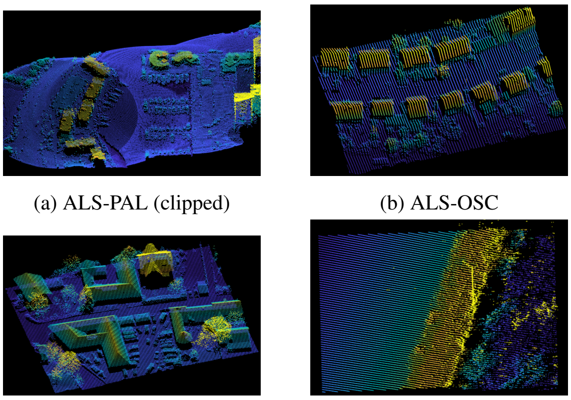
(a) Rotating Polyhedron

(b) Rotating Wedge

(c) Palmer Scanner

(d) Oscillating Mirror

Figure 1. Most common scan mechanics and their scan pattern on a horizontal planar surface; v indicating the direction of motion, LRF is the Laser Rang Finder element



(a) ALS-PAL (clipped)

(b) ALS-OSC

(c) ALS-ROT

(d) MSL-360

Figure 2. Height color coded 3D view of the test data

Label	Sensor name	Type	Points	Avg. point density of all echos $[pts/m^2]$	Median of inhomogeneity	Interpolation grid size $[m]$
ALS-PAL	Optech CZMIL Nova	Palmer scanner	66701	0.3^1	2.1	2.000
ALS-OSC	Leica ALS50 system	oscillating mirror	61112	5.5^2	5.8	0.500
ALS-ROT	Riegl VQ-880-G	rotating polyhedron	63108	6.3^2	1.2	0.250
MLS-360	Z+F Profiler 9012	rotating wedge	79785	3103.7	14.8	0.025

Table 1. Data overview; [1]only one of seven channels used, [2]single strip only

2.2 Framework

In this section the spatial selection framework and the supported queries are described. As mentioned before, nearest neighbour and region queries are the most common ones when processing 3D point clouds. Fixed distance queries, a term which is often found in literature, relates to a circular and spherical region in 2D and 3D, respectively. Hence, fixed distance queries are an isotropic subset of region queries as defined in the proposed framework. For the sake of completeness, it should be mentioned that circular regions correspond to infinite vertical cylinders in 3D. Since the framework also supports finite cylinders as search region, we use the terms circular region for infinite cylinders and cylindrical region for finite cylinders to avoid naming confusions. Furthermore, the framework supports 2D window and 3D box region queries as well. Such rectangular neighbour definitions are relevant for analysis over the full domain avoiding overlapping neighbourhoods.

Accordingly, the framework supports the following group of basic neighbourhood definitions:

- 2D and 3D k-nearest neighbour queries and
- region queries in 2D and 3D, such as, circle, sphere, cylinder, window and box regions

Whereas k-nearest neighbours can compensate varying point densities, unfavorable neighbour selection will occur for inhomogeneous point distributions. There, neighbours are not located symmetrical but rather on one side of the search point, which will lead to extrapolation artefacts. Increasing the number of neighbours k will also increase the chance of retrieving neighbours that fully surround the search point. However, a larger number of neighbours entails undesired smoothing. One method to overcome this issue is to introduce a quadrant or octant based selection of nearest neighbours representing a generation based neighbour definition. When searching, e.g., 8 neighbours in quadrant selection mode, the framework searches for the two closest points in each quadrant independently, as shown in Fig. 3. If one quadrant is not occupied, only 6 neighbours will be found, two empty quadrants results in 4 neighbours, etc.. Such an adapted kNN strategy avoids extrapolation with a small set of neighbours.

Beside the basic neighbourhood definitions, the proposed framework also supports advanced neighbour definitions, namely:

- the combination of basic neighbourhood definitions (intersection or union) and
- incremental queries; Starting from a minimal region, the search area is incrementally increased until a certain condition is fulfilled or the maximum search region is reached.

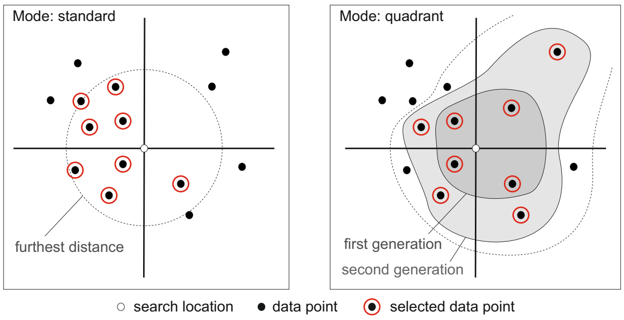


Figure 3. Different selection modes for querying eight nearest neighbours

The intersection of kNN and region queries is, e.g., useful to limit the kNN query to a maximum search distance in order to avoid bridging large gaps. From an implementation point of view, the maximum search distance can also be used to speed up the kNN algorithm, if the search dimension of both queries matches. It allows neglecting nodes of the spatial index if the distance between the node's bounding box and the search point exceeds the maximum search distance. The situation is different when joining kNN and region queries. Such union queries can be useful when the data set contains a significant number of small clusters with a lot of points that do not represent the true point density of the overall data set (e.g. caused by merging data from multiple sensors with largely different resolution). Hence, one might use the kNN query in general but wants to get all neighbours within the expected overall resolution. Thus, in locations close to or within clusters, all cluster points will be returned as neighbours. In contrast to the intersection case, direct optimisation of the different queries is not possible for the union case. In the worst case, the kNN and the region queries have to be performed separately and the resulting set of neighbours have to be merged. However, assuming a data set with individual point clusters as described above, the neighbours of the kNN query will usually exceed the region query, so the region query can be omitted. Only if all kNN neighbours are located within the search region, the region query needs to be performed and the result sets needs to be merged. Under this assumption the join query will achieve nearly the same performance as a standard kNN query.

The most complex spatial neighbourhood definition within the proposed framework are incremental queries. Starting from a minimal search region, the region is increased incrementally until a certain neighbour condition is fulfilled or the maximum search region is reached. The central idea of this query type is a neighbourhood definition 'as small as possible, but as large as necessary'. As stop criterion, e.g., the number of found neighbours or a minimum number of occupied quadrants or octants, is repetitively checked. In that sense it is similar to a kNN query with the corresponding selection mode (cf. Fig. 3), but based on an isotropic neighbourhood definition. As an efficient approximation of such an incremental query, we implemented this query type via a cell based approach. Starting from a minimum set of core cells, further cells around the core are added in each incremental pass with the cell size as discretisation interval. This implements a generation based neighbourhood definition as shown in Fig. 4.

As mentioned in Sec. 1.1, sampling strategies are used for reducing as well as for homogenising the data set. In the proposed framework, a cell based approach in 2D or 3D is utilized which further allows implementing an efficient incremental region query. The selection of a single or multiple representative points per cell can be controlled by three different methods (see Fig. 4):

- select all points per cell
- select a single point based on a certain condition like, min, max, n$^{\text{th}}$min, n$^{\text{th}}$max, mode, closest-to-centre, etc.
- compute an aggregated cell point located in the centre of gravity of all cell points based on an accumulative statistical feature such as mean, quantile, variance, rms, etc.

For feature computation either a certain coordinate (x, y, z) or an arbitrary attribute of the point can be used. Compared to traditional sampling strategies, the proposed framework computes the sampling on-the-fly and does not store the results in general. As it turns out, there is no single sampling strategy that is optimal for all different processing tasks. E.g., when interpolating a DSM the highest cell point is favorable, whereas for a DTM lower lying points are preferred. For other processing tasks attribute based selections might by more appropriate. E.g., for computing road segments, points with low amplitude are most relevant. In this case, one would select the point with the minimum amplitude in each cell. In case of large inhomogeneity, aggregative features can be beneficial to compute a stochastically independent point distribution. In case of very dense point clusters the assumption that each measurement is independent typically doesn't hold, especially for scanners based on the phase-shift principle. Averaging points in cells can result in more appropriate input for a subsequent optimisation algorithm.

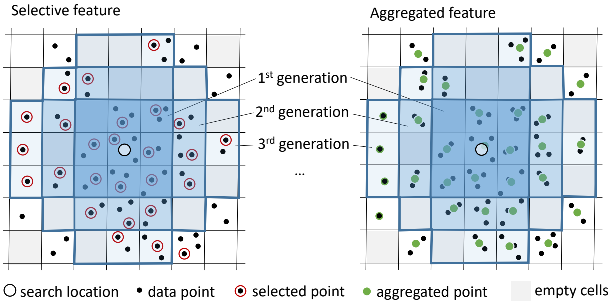


Figure 4. Sampling based on selective and aggregative features

In general, spatial queries do not return the found points in a specific order, except for kNN. Since sorting points by their distance to the search point is an integral part of any kNN algorithm, the returned neighbours are always sorted according to the used dimension. Hence, for a 2D kNN query points will be sorted by the 2D distance and for the 3D case by 3D distances, respectively. If the processing algorithm requires a specific order of the neighbours, it cannot rely on a distance based sorting since the neighbourhood definition is decoupled from the processing algorithm. To avoid unnecessary sorting operations, the proposed framework provides an additional option so sort the neighbours by distance (2D or 3D), coordinates or attributes. If the requested sorting is directly given by the spatial query, the additional sorting step is omitted.

The proposed framework does not only support constant values as parameters within the queries (k, region extents) but also features local adaptions based on the attributes of the current search point. Thus, the neighbourhood definition can be controlled by parameters like the local point density. This concept is inherently linked to the handling of point attributes. A detailed description of this feature is beyond the scope of this paper.

2.3 Evaluation

To demonstrate the flexibility and power of the proposed framework, the following typical processing tasks are performed for all test data sets: (i) interpolation of a regular grid using moving least-squares and (ii) computation of surface normal vectors including calculation of linearity and planarity derived via principle component analysis (PCA) of the 3D structure tensor.

To calculate the grids, the following three neighbourhood definitions and the grid size reported in Table 1 were used:

- standard kNN with 12 neighbours (label: kNN12)
- quadrant based kNN with 8 neighbours (label: quad$k$NN8)
- incremental circle query ($r = \{2, 4, 6, 8, 10\} * cellSize$) based on closest-to-centre sampling with a cell size of half the grid size (label: incCirc)

For visual inspection a hill shading of each grid model is computed. Hill shadings clearly reveal interpolation artefacts, because they depend on the first derivative of the surface.

For surface normal and planarity/linearity feature extraction the following two neighbourhood definitions are applied:

- a standard kNN with 16 neighbours (label: kNN16)
- kNN with 16 neighbours based on closest-to-centre sampling with a cell size that is half the grid size (label: samp$k$NN16)

A qualitative analysis of the results will be given in the next section.

3. RESULTS AND DISCUSSIONS

When visually inspecting the results, one has to keep in mind the different data set properties and the different scan patterns. ALS-ROT was captured with a scanner operating a rotating polyhedron, which results in a very homogeneous point cloud (f=1.2). The next data set, regarding homogeneity, is ALS-PAL acquired by a Palmer scanner (f=2.1). After that, ALS-OSC using an oscillating mirror is ranked (f=5.8). Finally, MLS-360 employing a 360° rotating wedge features the largest inhomogeneity (f=14.8). Hence, we expect ALS-OSC and MLS-360 to be problematic for processing. Please note that the presented point density and inhomogeneity factors are not generally applicable to the scanner models itself, since the point distribution is effected by the chosen scanner parameters, the travel velocity,

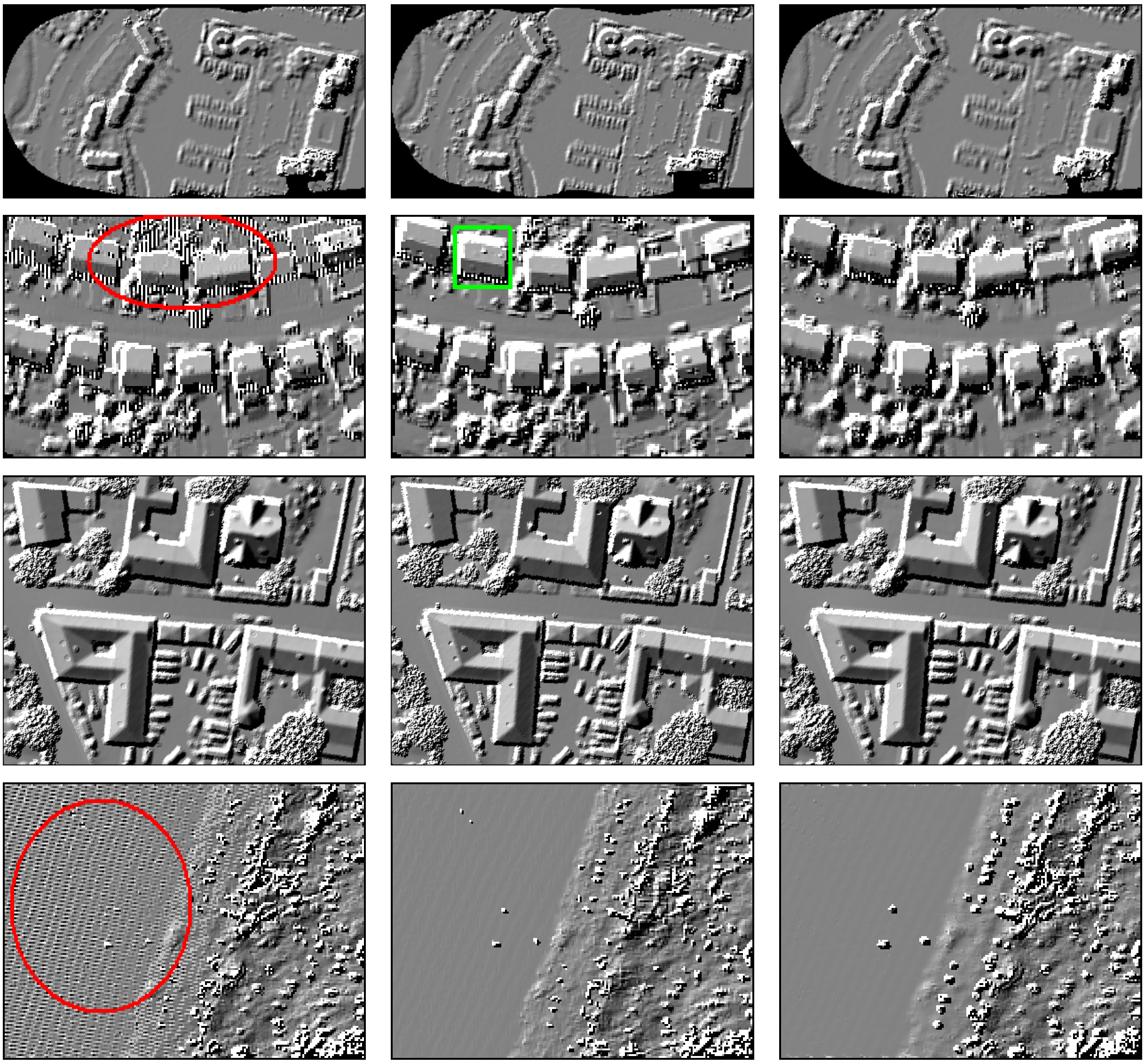


Figure 5. Shading results of data sets; Strong interpolation artefacts are visible in the left column marked by the red circle; Interpolation methods from left to right: kNN16, quad$k$NN8 and incCir; Data set from top to bottom: ALS-PAL, ALS-OSC, ALS-ROT and MLS-360

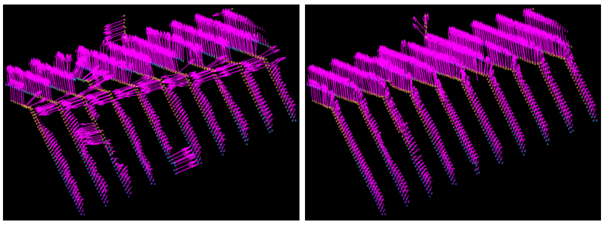


Figure 6. Normals of a roof of data set ALS-OSC. The selected house is marked by a green rectangle in Fig. 5. Left: kNN16, right: samp$k$NN16. Because of the large inhomogeneity poor normals will result for a standard kNN

measuring distance, multi channel or multi strip coverage, etc. Nevertheless, the scan pattern itself will stay unchanged.

The grid interpolation results are presented as hill shadings in Fig. 5. From left to right, the following neighbourhood definitions are used: kNN with 12 neighbours (kNN12), quadrant based kNN with 8 neighbours (quad$k$NN8), and incremental circular query (incCir) based on a closest-to-centre sampling (see Sec. 2.3 for further details). The red circles mark interpolation artefacts for kNN12 (left column) in data set ALS-OSC (row 2) and MLS-360 (row 4). The other two neighbourhood definitions show no, or very little artefacts. Fig. 5 shows that the surface models obtained by quad$k$NN8 (middle column) appear slightly sharper than the others (cf., ships in harbour, ridge lines of buildings, etc.). This can be attributed to the fact that only 8 compared to 12 neighbours are used. On the other hand, still a slight sinuous pattern is visible on the road of MLS-360 (bottom centre picture). This pattern practically vanishes using the incCir neighbourhood (bottom right picture).

The feature extraction results are presented in Fig. 6 and Fig. 7. The features were computed based on k-nearest neighbours ($k = 16$) with either standard kNN (kNN16) or closest-to-centre sampling (samp$k$NN16). The linearity and planarity features are shown in Fig. 7. For ALS-PAL (first row) and ALS-ROT (third row), all three neighbourhood definitions resulted in nearly the same hill shading. In contrast, differences are visible for the calculated linearity and planarity. The points in

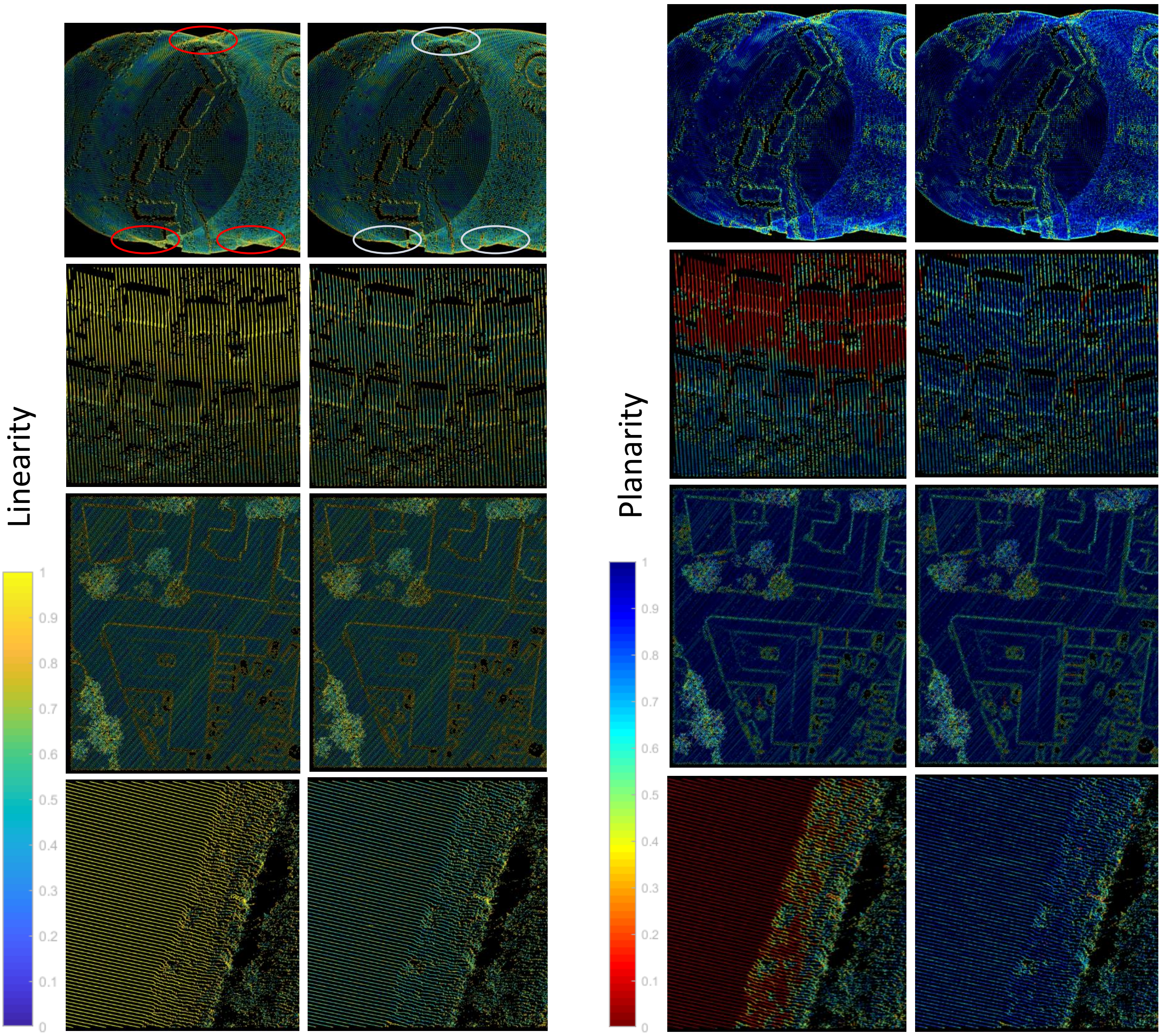


Figure 7. The linearity and planarity feature is highly influenced by the scan pattern if not compensated by the neighbourhood definition. This is visible in row 2 and 4. Left column: kNN16, right column: samp$k$NN16; Data set from top to bottom: ALS-PAL, ALS-OSC, ALS-ROT and MLS-360

the regions marked with red ellipses (top left picture) exhibit a high local linearity value. Identifying those regions within the hill shading, it can be seen that the surface is actually planar. Hence, the high linearity is an artefact of the overlapping circular scan pattern. In the top picture of the second column, the advanced neighbourhood definition (samp$k$NN16) was used and there, the points in the same regions exhibit much lower linearity values. For both features, the influence of the point distribution is pronounced for ALS-OSC (second row) and MLS-360 (fourth row). The upper (northern) half of ALS-OSC features high linearity and low planarity values using kNN16 although most of the areas are planar. It can be seen that samp$k$NN16 can largely compensate the influence of the scan pattern. The same situation arises in the MLS-360 data set (fourth row). The north western part of the test area is a road. Consequently, the points should have low linearity and high planarity values. But again, this only holds for samp$k$NN16.

Finally, we demonstrate that also normal vectors are affected by inhomogeneous point distributions. Fig. 6 shows a 3D view including normal vectors of a roof. The selected roof is marked with a green rectangle in Fig. 5. The normal vectors were derived by kNN16 (left) and samp$k$NN16 (right) neighbourhood. The kNN16 neighbourhood leads to incorrect normals for a considerable number of points, although the 3D points are nicely aligned with the roof surface. In contrast, using samp$k$NN16 neighbourhood of the proposed framework, correct normal vectors can be derived.

4. CONCLUSIONS

In this manuscript a spatial search framework is proposed that supports a variety of different atomic search strategies in 2D and 3D. Furthermore, those basic search strategies can be extended and flexibly combined to sophisticated neighbourhood definitions. In conjunction with a powerful cell sampling concept, even inhomogeneous 3D point clouds from many different sensors or acquisition techniques, respectively, can be processed in an appropriate manner.

Based on an inhomogeneity measure, the isotropy of the point distribution of four different test data sets were quantified. In general, it can be stated that for homogeneous point distributions the choice of the neighbour definition is uncritical. However, the larger the inhomogeneity the more computational artefacts can be observed. Such data sets require carefully chosen neighbourhood definitions. For the typical applications of surface interpolation and feature extraction we could demonstrate that the framework can cope with demanding data sets where standard neighbourhood definition fail to achieve satisfactory results. As shown, feature extraction is more sensitive to inhomogeneous data than surface interpolation.

The problem of inhomogeneous point distribution at strip borders for ALS scanners using oscillating mirrors is well known. Often those border points are simply dropped to avoid problems within the processing pipeline. Nevertheless, those edge points do contain valuable information which is, e.g., of particular relevance for quality control and strip adjustment. With the use of the proposed framework, stripping of those points can be omitted and their information is preserved.

The investigations have also revealed a shortcoming of the current cell based sampling approach. If the grid lines of the sampling structure are nearly parallel to the scan lines of inhomogeneous data (as is the case for ALS-OSC), moiré patterns can be observed. Those patterns can be seen in the linearity and planarity feature of ALS-OSC (second row of Fig. 7) using the sampkNN16 neighbourhood (second and fourth column). To resolve this issue, either the data or the sampling grid structure need to be temporarily rotated by e.g. 45°. In future work, the framework should be also evaluated on additional data sets, like single photon LiDAR and photogrammetric point clouds. While dense image matching of nadir images results in a very homogeneous point distribution, point clouds derived from oblique images are of specific interest in this context.

To sum up the central idea of the framework: By decoupling data processing algorithms from the actual neighbourhood definition and spatial neighbour selection, new processing chains can be adapted faster and the complexity of the program code stays low. As such, it facilitates point cloud processing and enables scientists to concentrate on the actual research problem rather than keeping the impact of unfavourable point distributions on the final results under control.

ACKNOWLEDGEMENTS

Norbert Pfeifer and Wilfried Karel were partially supported by the EU Marie Sklodowska-Curie RISE project VOLTA. The authors thank the United States Army Corps of Engineers, the City of Vienna, Municipal Department 41, and company Riegl for providing the scan data for this manuscript.

REFERENCES

Blaszczak-Bak, W., Janicka, J., Suchocki, C., Masiero, A., Sobieraj-Żlobinska, A., 2020. Down-Sampling of Large LiDAR Dataset in the Context of Off-Road Objects Extraction. *Geosciences*, 10(6). https://www.mdpi.com/2076-3263/10/6/219.

Bobach, T., Umlauf, G., 2006. Natural neighbor interpolation and order of continuity.

CloudCompare Development Team, 2021. CloudCompare, 3D point cloud and mesh processing software . CloudCompare[GPL] Software. www.cloudcompare.org (4 February 2021).

de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O. C., 2000. *Delaunay Triangulations*. Springer Berlin Heidelberg, Berlin, Heidelberg, 183–210.

Devillers, O., Pion, S., Teillaud, M., Logiciel, T., Prisme, P., 2001. Walking in a Triangulation. *International Journal of Foundations of Computer Science*.

Filin, S., Pfeifer, N., 2005. Neighborhood Systems for Airborne Laser Data. *Photogrammetric Engineering & Remote Sensing*, 71, 743-755.

Glira, P., Pfeifer, N., Briese, C., Ressl, C., 2015. A Correspondence Framework for ALS Strip Adjustments based on Variants of the ICP Algorithm. *Photogrammetrie - Fernerkundung - Geoinformation*, 2015(4), 275-289. http://dx.doi.org/10.1127/pfg/2015/0270.

Held, P., Schneider von Deimling, J., 2019. New Feature Classes for Acoustic Habitat Mapping—A Multibeam Echosounder Point Cloud Analysis for Mapping Submerged Aquatic Vegetation (SAV). *Geosciences*, 9(5). https://www.mdpi.com/2076-3263/9/5/235.

Isenburg, M., Liu, Y., Shewchuk, J., Snoeyink, J., 2006. Streaming Computation of Delaunay Triangulations. *ACM Trans. Graph.*, 25(3), 1049–1056. htps://doi.org/10.1145/1141911.1141992.

MeshLab Development Team, 2021. MeshLab: an Open-Source Mesh Processing Tool. MeshLab Software. www.meshlab.net (4 February 2021).

Niemeyer, J., Rottensteiner, F., Soergel, U., 2012. Conditional Random Fields for LIDAR Point Cloud Classification in Complex Urban Areas. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1, 263-268.

Otepka, J., Ghuffar, S., Waldhauser, C., Hochreiter, R., Pfeifer, N., 2013. Georeferenced Point Clouds: A Survey of Features and Point Cloud Management. *ISPRS International Journal of Geo-Information*, 2(4), 1038–1065. https://www.mdpi.com/2220-9964/2/4/1038.

Pfeifer, N., Falkner, J., Bayr, A., Eysn, L., Ressl, C., 2021. Test Charts for Evaluating Imaging and Point Cloud Quality of Mobile Mapping Systems for Urban Street Space Acquisition. *Remote Sensing*, 13(2). https://www.mdpi.com/2072-4292/13/2/237.

Puttonen, E., Lehtomäki, M., Kaartinen, H., Zhu, L., Kukko, A., Jaakkola, A., 2013. Improved Sampling for Terrestrial and Mobile Laser Scanner Point Cloud Data. *Remote Sensing*, 5(4), 1754–1773. https://www.mdpi.com/2072-4292/5/4/1754.

Sankaranarayanan, J., Samet, H., Varshney, A., 2007. A fast all nearest neighbor algorithm for applications involving large point-clouds. *Comput. Graph.*, 31(2), 157–174. http://dx.doi.org/10.1016/j.cag.2006.11.011.

Schaer, P., Skaloud, J., 2010. Optimizing Computational Performance for Real-Time Mapping with Airborne Laser Scanning. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 38.

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105, 286 - 304.